Dr. Hans-Joachim Pross

(E-Mail: hajo.pross@de.ibm.com)
ist Field Technical Professional bei der IBM in Köln. Darüber hinaus ist er Community of Practice Lead für Change- und Configuration-Management. Nach seinem Studium der Physik arbeitete er zunächst beim Bank-Verlag als DV-Koordinator, war dann lange Jahre als Consultant bei Softwab tätig, ehe er vor mehr als 9 Jahren zu IBM Rational wechselte. Dort ist er seither vertriebsunterstützend für Rational Produkte tätig. Seine aktuellen Schwerpunkte bilden die Beratung für Lösungen im Bereich Änderungs- und Konfigurationsmanagement, Asset-Management und natürlich Cloud Computing.

Marco Lüscher

(E-Mail: marco.luescher@ch.ibm.com)
ist Field Technical Professional bei der IBM in Zürich. Er ist seit 1986 in der IT tätig, ausgebildet als Analytiker/Programmierer, danach als Freiberufler unterwegs, stieß er im Jahr 2000 zu Rational Software. Anfangs im Bereich Softwaretesting tätig, verlegte er seinen Schwerpunkt in Richtung Software Change- und Release-Management. Heute ist er vor allem in den Bereichen CALM und Enterprise Modernization tätig und bei Rational (CH und A) Ansprechpartner für Cloud.

Flexiblere Softwareentwicklung und Cloud – wie fängt man an?

Die Herausforderungen in der Softwareentwicklung sind bekannt und stets dieselben: Beste Ergebnisse in möglichst kurzer Zeit mit verteilten Teams bei schrumpfenden Budgets sind gefordert. Die Lösungsansätze sind teils nicht neu: flexiblere, agile Vorgehensmodelle oder dynamische, sich dem Bedarf anpassende Teamstrukturen. Ist es da nicht konsequent, auch die benötigten Entwicklungsinfrastrukturen zu flexibilisieren? Cloud Computing als Bereitstellungsmodell ermöglicht gerade im Softwareentwicklungs- und Testumfeld großes Verbesserungspotenzial durch das schnelle und automatisierte zur Verfügung stellen von Systemen. Wir zeigen auf, wie Kunden und IBM die Cloud nutzen.

Cloud Computing – was ist das überhaupt?

Inzwischen dürfte allgemein anerkannt sein (vgl. z. B. [OBJ10]), dass Cloud Computing mehr als nur ein weiterer Hype ist. Dies wird z. B. in Google trends [Goo] durch eine seit 2008 beständig ansteigende Anzahl von Suchen nach dem Schlagwort „Cloud Computing“ eindrucksvoll bestätigt (siehe Abb. 1).

Bei Terminen zu Cloud Computing wird man trotzdem oft mit der an sich simplen Frage konfrontiert, was Cloud Computing denn überhaupt ist. Und wenn bei Terminen diese Frage einmal nicht gestellt wird, stellt man sie am besten selbst, da man sonst im Verlaufe von Gesprächen häufig feststellt, dass inzwischen zwar jeder eine Vorstellung von Cloud Computing hat, diese Vorstellungen jedoch mitunter sehr unterschiedlich sind. Gern wird z. B. Cloud Computing mit einfacher Virtualisierung gleichgesetzt. Dabei gibt es bereits eine stabile Definition vom National Institute of Standards and Technology [Nis09] sowie eine hierauf basierende von Forrester [For]: *Cloud Computing stellt eine standardisierte IT-Lösung (Service, Software oder Infrastruktur) über Internet-Technologien auf Selbstbedienungsbasis mit nutzungsabhängiger Bezahlung zur Verfügung* (siehe Abb. 2).

Genau diese Merkmale sind es, die Cloud Computing, z. B. für die Softwareentwicklung, als interessante Alternative oder Erweiterung zu bestehenden Lösungen erscheinen lässt.

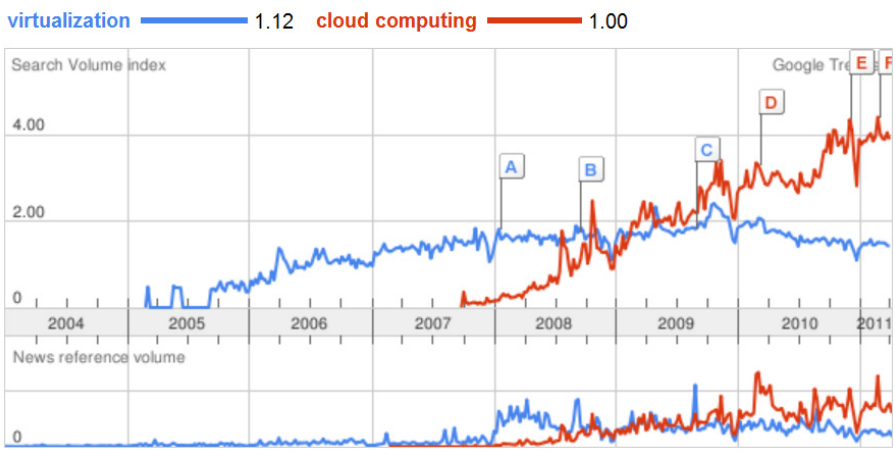

Abb. 1: Cloud Computing, längst mehr als ein weiterer Hype

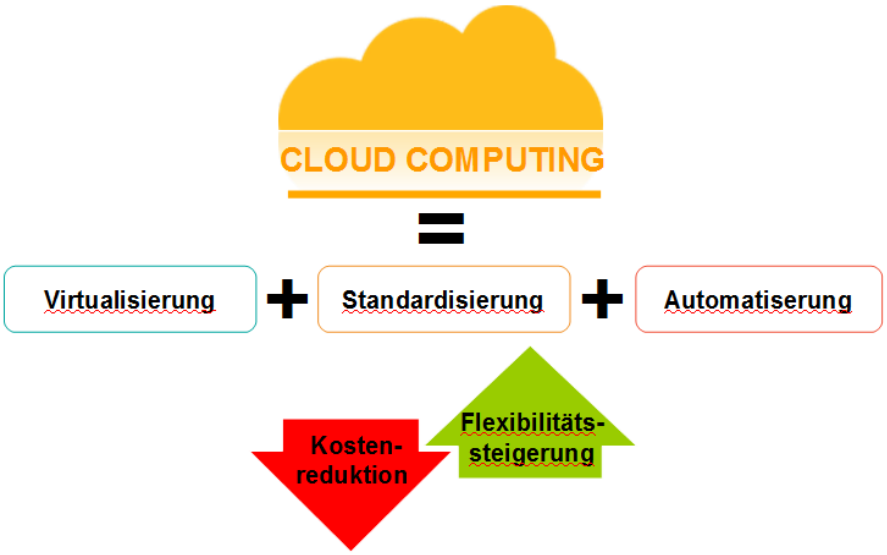

Abb. 2: Schematische Darstellung von Cloud Computing

- *Nutzungsabhängige Bezahlung:* Die Kosten einer Cloud-basierten IT-Lösung sind proportional zu deren Nutzung. Anfängliche Investitionen entfallen bei der Nutzung von Public Clouds gänzlich und werden stattdessen in operationale Kosten umgewandelt. Für Unternehmen mit eigener Cloud-Infrastruktur entfallen die Anfangsinvestitionen zwar nicht, diese sollten – bezogen auf ein Projekt – jedoch niedriger ausfallen als bei etablierten Lösungsansätzen, da nicht mehr jedes einzelne System auf dessen individuelle Maximallast ausgelegt sein muss. Typischerweise sind die Auslastungsprofile mehrerer Projekte unterschiedlich, sodass Spitzen der Nutzung des einen Projektes in Zeiten geringerer Auslastung der anderen Projekte anfallen.
- *Skalierende Lösungen:* Durch die dem Cloud Computing zugrunde liegende Virtualisierung können Ressourcen dynamisch bedarfsgerecht umverteilt werden. Eine solche Ressourcenumverteilung kann sowohl automatisch erfolgen, als auch manuell durch das Projektteam angestoßen werden.
- *Automatisierte Bereitstellung:* Die Bereitstellung einer IT-Lösung erfolgt automatisiert und wird durch den Nutzer selbst veranlasst. Zur Bereitstellung gehört natürlich auch das Zurückgeben von Ressourcen sowie deren dynamische Vergrößerung oder Verkleinerung. Hier wird ein entscheidender Unterschied zur Virtualisierung deutlich, die inzwischen zwar weite Verbreitung gefunden hat, jedoch nicht über diesen Grad an Automatisierung und Selbstbedienung verfügt.
- *Standardisierte IT-Lösung:* Voraussetzung für die Selbstbedienung ist, dass eine Anzahl vorgefertigter standardisierter Lösungen in einem „Kaufhaus“ bereitgestellt wird, aus der der Anwender das für ihn passende Angebot auswählt. Selbstverständlich kann die eigene Instanz anschließend individuell angepasst werden.

Möglichst schnell, möglichst gut und möglichst preiswert

Wer kennt nicht die typischen Herausforderungen, nach denen man bei sinkenden Budgets mit gleichbleibender oder steigender Qualität bei sinkendem „Time-to-Market“ seine Entwicklungsprojekte durchzuführen hat. Genau für diese Anforderungen bieten die Versprechungen des Cloud Computings mögliche Lösungsangebote:

- 85 % aller Computerkapazitäten laufen „idle“, 30-50 % aller Server sind für Testzwecke reserviert, wobei diese dann oft mit weniger als 10 % der Zeit genutzt werden. 70 % der Kosten entfallen auf die Wartung der bestehenden IT-Infrastruktur [IDM08]. Eine *nutzungsabhängige Bezahlung* sowie eine *skalierende Lösung*, die nicht auf die jeweilige Maximallast auszulegen ist, bergen hier erhebliches Einsparpotenzial.
- 30 % aller Störungen in der Produktion werden durch Tests auf falsch konfigurierten Umgebungen verursacht. Testressourcen weisen häufig eine unbekannte Konfiguration auf. *Standardisierte IT-Lösungen* für Testressourcen garantieren gleichbleibende Testumgebungen.
- Verzögerte Tests gelten als größter Faktor für verzögertes „Time-to-Market“. Vorhandene Testressourcen sind schwer zu finden. Dynamische Teams werden durch lange Bereitstellungszeiten oftmals „ausgebremst“. Die *automatisierte Bereitstellung* ermöglicht, Entwicklungsumgebungen und Testressourcen nach Bedarf genau zum gewünschten Zeitpunkt zu erstellen und wieder freizugeben.

Low hanging fruits?

Der einfachste und schnellste Weg in die Cloud besteht darin, sich einmal mit Public Cloud-Angeboten auseinanderzusetzen. Meist ist lediglich ein Vertrag mit einem Cloud-Anbieter zu schließen, um starten zu können. Die entstehenden Kosten sind typischerweise proportional zur Nutzung und entstehen im Allgemeinen erst dann, wenn tatsächlich Instanzen erstellt und genutzt werden.

Von der Seite der Anwendungen im Softwareentwicklungsumfeld drängen sich solche auf, die für begrenzte Zeiträume benötigt werden: Lasttests mit einer großen Zahl von simulierten Nutzern, funktionale Tests oder Builds. Auch Server für Projekte mit einer begrenzten Laufzeit fallen unter diese Gruppe.

Ist bereits eine eigene Cloud-Infrastruktur vorhanden, können die Geschwindigkeits-, Qualitäts- oder Selbstbedienungsvorteile der Cloud für sämtliche Bereiche der Softwareentwicklung genutzt werden. Der Vorteil ist umso größer, je schnelllebiger ein Projekt ist bzw. je häufiger bestimmte Tasks wiederholt werden.

Neben diesen beiden „reinen“ Cloud-Typen ist auch eine Hybrid-Lösung nicht uninteressant, bei der man die wesentlichen Teile auf der eigenen Cloud-Infrastruktur abbildet und lediglich besondere Lastspitzen auf die Systeme eines Cloud-Anbieters auslagert.

Lasttest aus der Cloud

Lasttests werden in vielen Projekten häufig mit einer überschaubaren Anzahl „virtueller Tester“ auf realer Hardware durchgeführt. Wenn überhaupt, dann seltener und nur für einen kurzen Zeitraum, kommen große Zahlen von „virtuellen Testern“ zum Einsatz. Cloud Computing auf

einer Public Cloud bietet hier die Chance, die Limitierung der bestehenden Hardware durch dynamisch provisionierte virtuelle Test-User auf virtuellen Computern automatisiert zu erzeugen und nach den Tests wieder zu deprovisionieren.

Die Kosten für Hard- und Software fallen nach dem „Pay-as-You-Use-Prinzip“ an. Ein Angebot der IBM basierend auf dem IBM Rational Performance Tester und der „IBM Smart Business Development and Test on the IBM Cloud“ ist in Vorbereitung und befindet sich zurzeit (Stand April 2011) im Beta-Stadium.

Wie in Abbildung 3 dargestellt, werden die zu testenden Systeme vom Rational Performance Tester über virtuelle Tester auf realer Hardware im Unternehmen angesprochen. Die Konfiguration erzeugt normale Last auf den zu testenden Servern.

Darüber hinaus werden durch den Rational Performance Tester automatisiert virtuelle Rechner in der IBM Smart Business Development and Test-Cloud erzeugt, die bereits konfigurierte Rational Performance Tester-Agenten enthalten. Diese Instanzen mit ihren „virtuellen Testern“ erzeugen, gesteuert durch den realen Rational Performance Tester, zusätzliche Last auf den zu testenden Applikationen. Nach Beendigung des Lasttests werden die virtuellen Rechner wieder entfernt und somit die Ressourcen zurückgegeben, sodass auch keine weiteren Kosten anfallen.

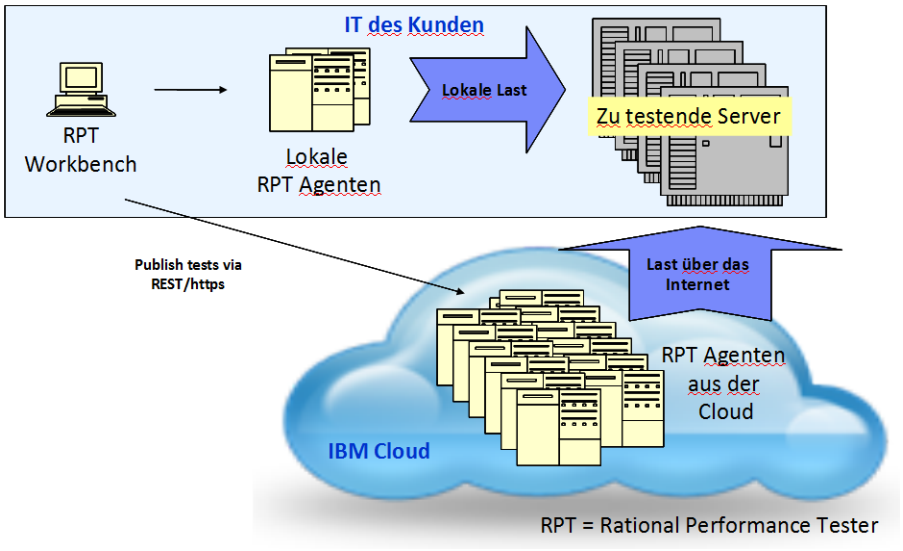

Abb. 3: Schematische Darstellung von Lasttests aus der Cloud

Software-Builds in der Cloud

Builds weisen mitunter einen erheblichen Bedarf an Ressourcen auf – sei es durch die immense Größe, der zu erzeugenden Software oder durch die Tatsache, dass eine erhebliche Zahl von unterschiedlich konfigurierten Buildumgebungen entsprechend den unterstützten Zielplattformen vorgehalten werden muss.

Lange Buildzeiten können dadurch adressiert werden, dass in der Cloud leistungsstärkere Hardware bereitsteht oder Builds auf zusätzliche Instanzen in der Cloud verteilt werden können.

Dieses Vorgehen spielt seine Vorteile dann am besten aus, wenn die virtuellen Instanzen bereits vorhanden sind oder die Zeit zur Aktivierung der Instanz im Vergleich zur Builddauer eine untergeordnete Rolle spielt. Werden die Buildzeiten durch eine große Zahl von wartenden Buildjobs in der Queue verursacht, hilft das in Abbildung 4 skizzierte Verfahren, bei dem die Buildsteuerung zusätzliche Buildinstanzen anfordert.

Wenn die Kombinatorik der unterstützten Zielplattformen (Datenbanken, Middleware, Browser, Sprachen, …) „zuschlägt“, ist ein verbreitetes Verfahren, nicht bei jedem Durchlauf alle Variationen zu erzeugen. Hier kann mittels geeigneter Buildsteuerung automatisiert dafür gesorgt werden, dass die Instanzen mit den gewünschten Zielumgebungen für den Build erzeugt werden, dieser darauf angestoßen wird und die Instanzen nach erfolgtem Build wieder entfernt werden. Denkbar ist auch, dass Instanzen zwar speziell für einen Build erzeugt werden, jedoch nicht unmittelbar nach diesem wieder entfernt werden, da häufig im Anschluss an einen Build eine Wiederholung oder ein Test erfolgt, der dieselbe Konfiguration benötigt.

Entwicklungsumgebung in der Cloud

Die Bereitstellung einer Entwicklungsumgebung für ein neues Projektteam beginnt herkömmlich mit der Beschaffung von Servern und der anschließenden Installation der Softwarewerkzeuge, was typischerweise einen Zeitraum von wenigen Wo-

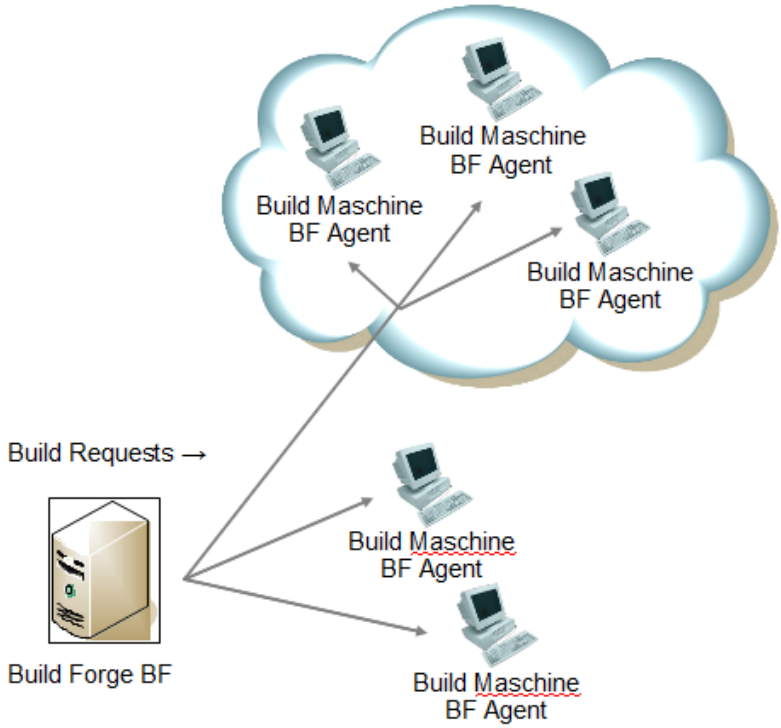

Abb. 4: Overflow Builds in der Cloud

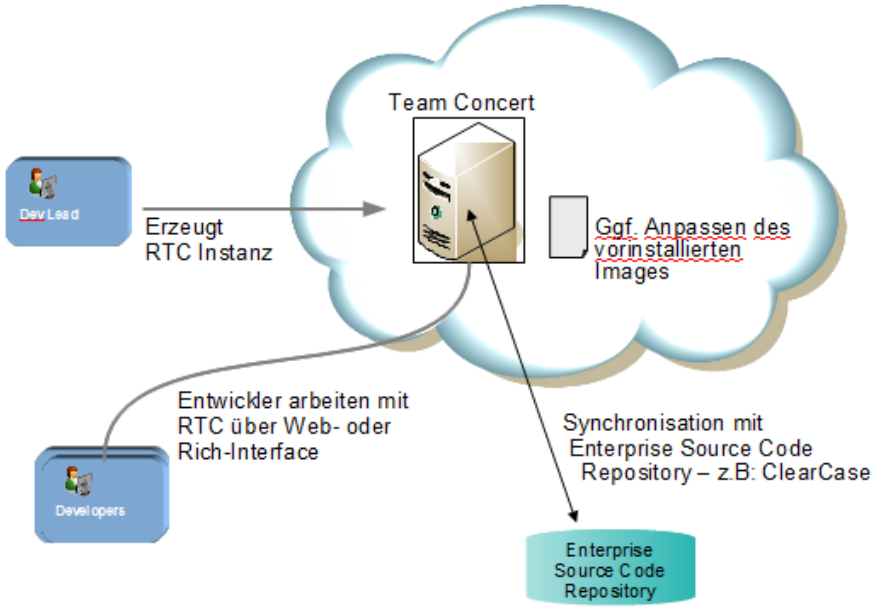

Abb. 5: Schematische Darstellung einer Team Concert-Entwicklungsumgebung in der Cloud

chen bis zu einigen Monaten in Anspruch nehmen kann. Durch Nutzung einer privaten Cloud-Infrastruktur oder einer öffentlichen Cloud mit vorinstallierten Images kann dieser Prozess im Idealfall auf weniger als eine Stunde reduziert werden.

Wie in Abbildung 5 schematisch dargestellt, erzeugt ein Projektleiter eine Instanz mit einem installierten und vorkonfigurierten Rational Team Concert-Server. Falls gewünscht, nutzt er die Synchronizer-Technologie, um Daten im Rational Team Concert mit Daten im Enterprise Source Code Repository abzugleichen.

Nachdem nun noch die Teameinladung versendet und von den Teammitgliedern akzeptiert wurde, ist das Team startbereit. In der „IBM Smart Business Development and Test"-Cloud stehen für diese Szenarien neben Images mit Rational Team Concert weitere Produkte der Jazz-Familie zur Verfügung.

Praxisbericht 1

Beispiel einer Private Desktop Cloud-Umgebung: Geografisch verteilte Entwicklung auf virtualisierten Desktops

Eine große Versicherungsgesellschaft beabsichtigte eine länderspezifische Versicherungslösung auszubauen und für ganz Europa wiederzuverwenden. Zu diesem Zweck wurden verschiedene Möglichkeiten in Betracht gezogen, wo und wie die Lösung entwickelt werden könnte.

Die bestehende Lösung basiert auf folgenden Komponenten:

- Cobol-Applikationen auf IBM i (ehemals AS/400)
- Tuxedo Middleware
- Java Fat Client
- Web Front-End.

Folgende verbindliche Anforderungen bestanden und mussten berücksichtigt werden: verteilte Entwicklungsteams, zentrales Rechenzentrum, Source Code sollte in der Schweiz bleiben, die Teams sind in verschiedenen Ländern tätig. Rollenbasierte Prozesse für das Gatekeeping (der Gatekeeper behandelt alle einkommenden Requests, beurteilt diese und reicht sie z. B. an den Change Management-Prozess

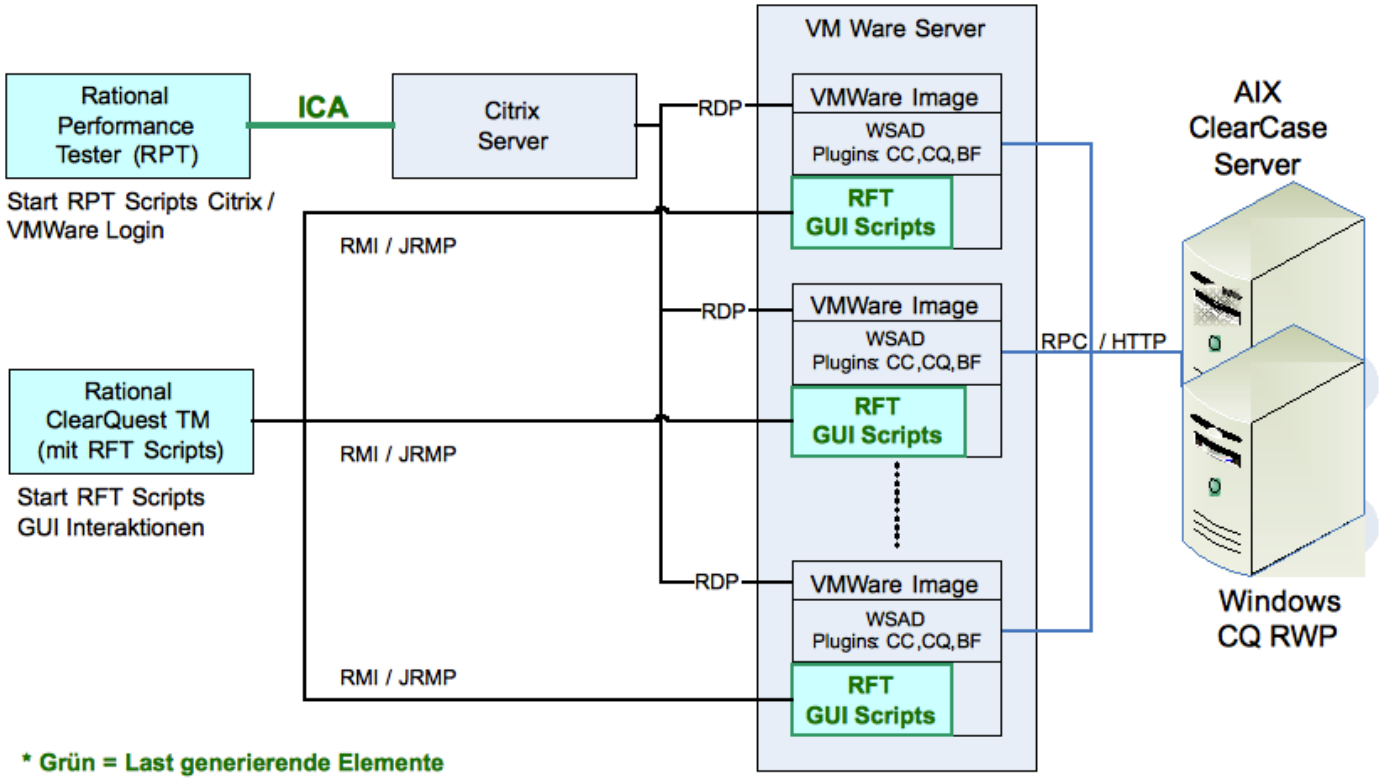

Abb. 6: Vertikaler Prototyp

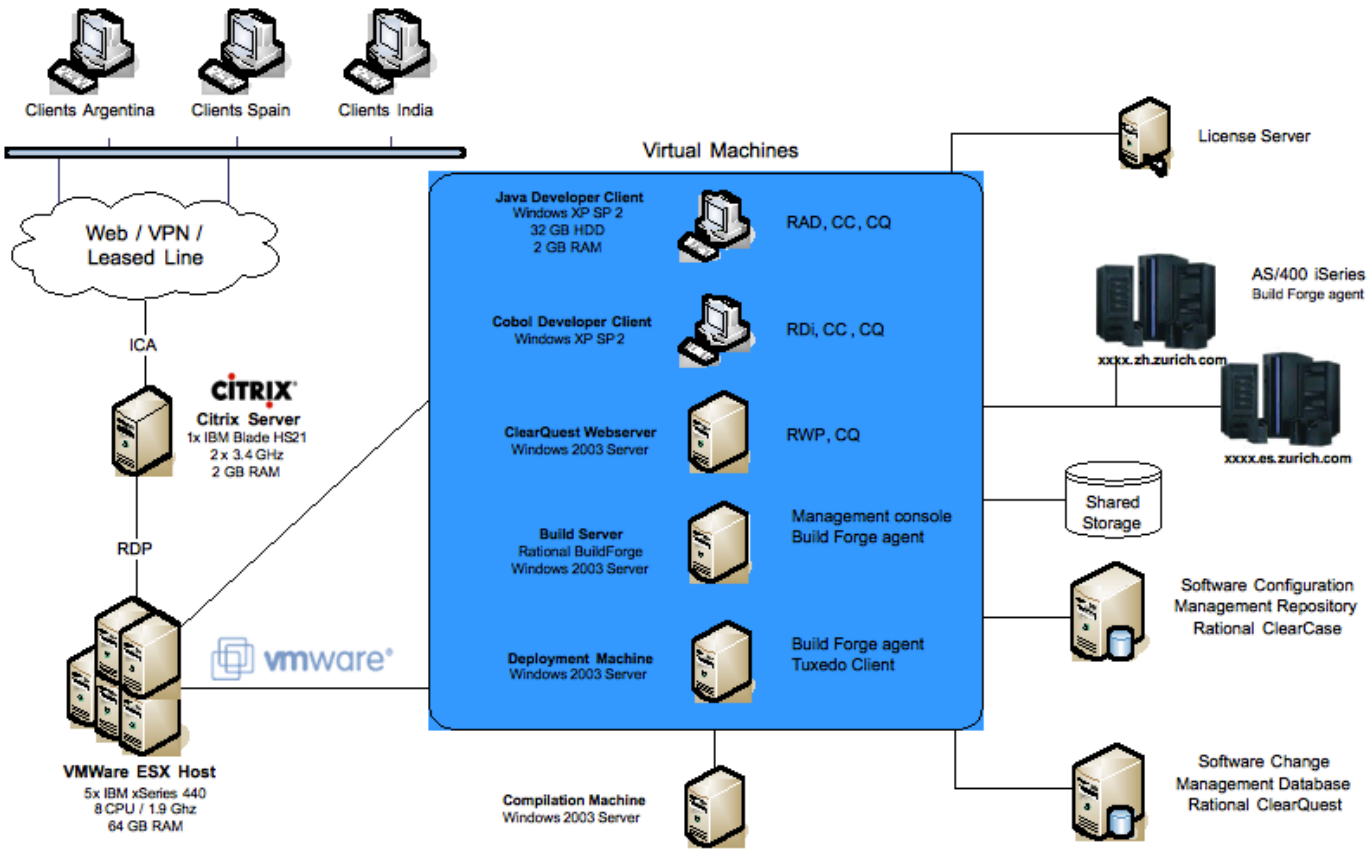

Abb. 7: Infrastruktur-Layout

weiter) waren ebenso zu beachten und für die Entwickler sollte eine einheitliche Umgebung zur Verfügung stehen.

Als Ziel wurde eine skalierbare Infrastruktur für bis zu 320 Entwickler vorgegeben.

Die bestehende Entwicklungsumgebung des Kunden musste zwingend berücksichtigt werden. Für die Disziplin Software Change- und Configuration-Management wurden Rational ClearQuest und Rational ClearCase eingesetzt. Als IDE dienten die Eclipse-basierten Tools Rational Developer for System i und Rational Application Developer. Für den Zugriff auf die IBM i wurde PCOMM als 5250 Emulator benutzt. Um die Builds zu erstellen, zu paketieren und auf die gewünschten Plattformen zu deployen, wurde Rational Build Forge eingesetzt.

Nach diversen Gesprächen mit internen und externen Architekten wurde als Lösung eine virtualisierte Client-Umgebung angestrebt. Da der Kunde einerseits eine Citrix und andererseits eine VMWare ESX Umgebung besaß, konnte man diese Infrastruktur nutzen und darauf eine Desktop Cloud aufbauen.

Für die Umsetzung wurde als erstes ein vertikaler Prototyp erstellt, mit welchem die gesamte Infrastruktur inklusive Netzwerk getestet werden sollte.

Der Ansatz mit Citrix wurde gewählt, um das Netzwerk effizienter zu nutzen. Für einen virtuellen Desktop benötigte Citrix nur etwa die Hälfte der Bandbreite vom Remote Desktop Protokoll (RDP). Dies ermöglichte eine hohe Auflösung auf der Client-Seite und eine flüssige Mausbedienung ohne „hüpfen“. Zudem konnte sichergestellt werden, dass die gemieteten Bandbreiten sicher ausreichen würden, um den zusätzlichem Netzwerkverkehr aufzufangen.

Nach diversen Tests entschied man sich für diesen Ansatz, wollte jedoch die Infrastruktur einer vertieften Performance-Analyse unterziehen. Die Hardware-Infrastruktur wurde so gewählt, dass man pro Server (IBM x440 Server) 64 Clients virtualisieren konnte. Physisch war jeder Server mit 64 GB RAM und 8 CPU's 1.9 Ghz bestückt. Abbildung 7 zeigt das gesamte Layout der Infrastruktur.

Die Testszenarien wurden an die täglichen Tätigkeiten der Entwickler angelehnt, entsprechende Use Cases wurden definiert, automatisiert und schlussendlich ausgeführt. Während der Testausführung konnte man die einzelnen Komponenten wie Citrix, VMware oder ClearCase Server und die Auslastung des Netzwerks überwachen. Die Testergebnisse bezüglich Performance waren durchweg positiv.

Schlussendlich wurde ein definitives Client Desktop Template zur Provisionierung der Clients erstellt. Durchschnittlich dauert die Erstellung eines Desktops circa 10 Minuten, danach erhält der Entwickler eine Mail mit den nötigen Verbindungsinformationen in Form einer URL mit ID und Initialpasswort und los geht's.

Zusammengefasst kann man sagen, dass die Hürde für Entwickler, eine virtuelle Desktopplattform zu nutzen, relativ groß war, da sie zwingend auf eine Netzverbindung angewiesen sind. Die guten Antwortzeiten, die stabile Umgebung und das schnelle On-Boarding neuer Teammitglieder konnte diesen Nachteil gut kompensieren. Potenzial zu einer „richtigen“ Cloud-Lösung besteht in den Bereichen Automatisierung/Self Service, Billing und Single Sign-on.

Praxisbericht 2

Wie nutzen IBM interne Teams die Public Cloud-Umgebung?

Sei es für Produktdemos innerhalb der Softwaregruppe, interne Softwareprojekte bei Operations, Performance Testing-Aktivitäten oder Projekten für Kunden – die IBM Smart Cloud [ibm] deckt all diese Bedürfnisse ab.

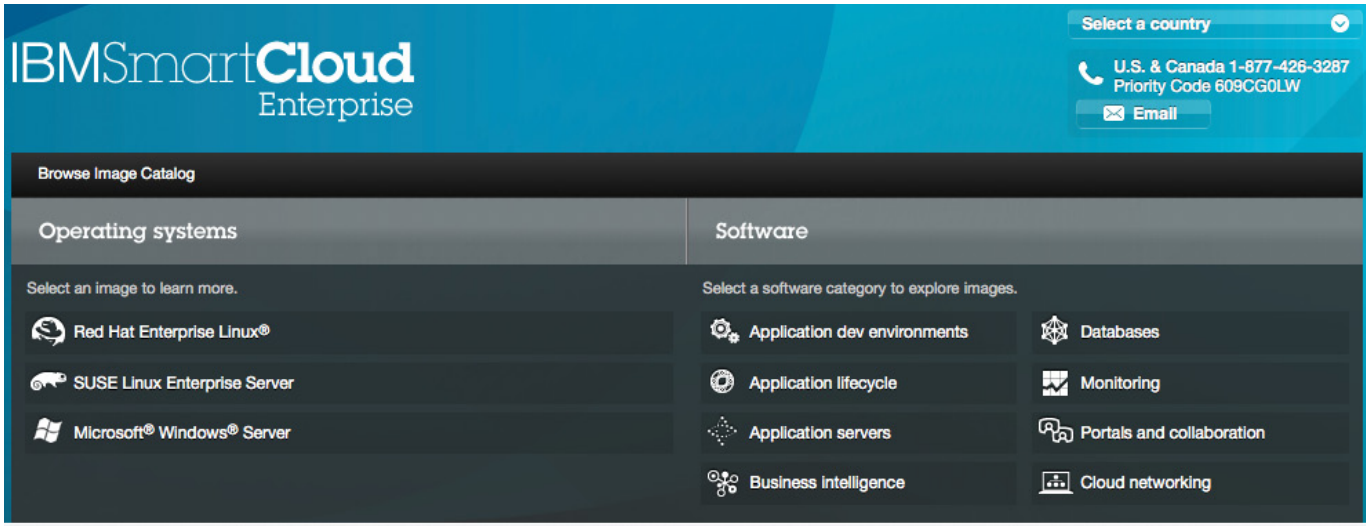

Abb. 8: IBM Smart Cloud Enterprise

Unsere Vertriebsorganisationen weltweit müssen sehr flexibel und schnell agieren können. Um zum Beispiel einen komplexen Proof of Concept durchführen zu können, wird eine entsprechende Infrastruktur benötigt – entweder beim Kunden oder bei IBM. Da ein definitives „Sizing" für ein allfälliges Deployment zu diesem Zeitpunkt noch schwer abzuschätzen ist, kann man meistens auch die benötigte Hardware und Software nicht abschließend bestimmen. Was liegt da näher als eine Cloud-Umgebung zu nutzen? Der Kunde muss keine spezielle Infrastruktur beschaffen, sondern wir nutzen gemeinsam die IBM-Cloud.

Der Vorteil liegt darin, dass wir gemeinsam die Plattform nutzen können und zu diesem Zeitpunkt keine großen Kosten anfallen.

Die Labs wiederum haben andere Anforderungen. Um schnell und kostengünstig eine Runtime-Plattform mit z. B. WebSphere Application Server und DB2 auf jeweils Windows und Linux zur Verfügung zu haben, eignet sich die Cloud bestens. So können einzelne Entwickler oder ganze Teams auf realistischen Umgebungen ihre Applikationen testen.

Damit für Load- und Performance-Tests nicht ganze Farmen von Servern und Agents installiert und konfiguriert werden müssen, verwenden die Tester auch die Cloud. So werden physische und virtuelle Server in größeren Szenarien kombiniert und je nach Bedarf über den Globus verteilt. Dies ermöglicht komplexe und realistische Runs in verteilten Umgebungen.

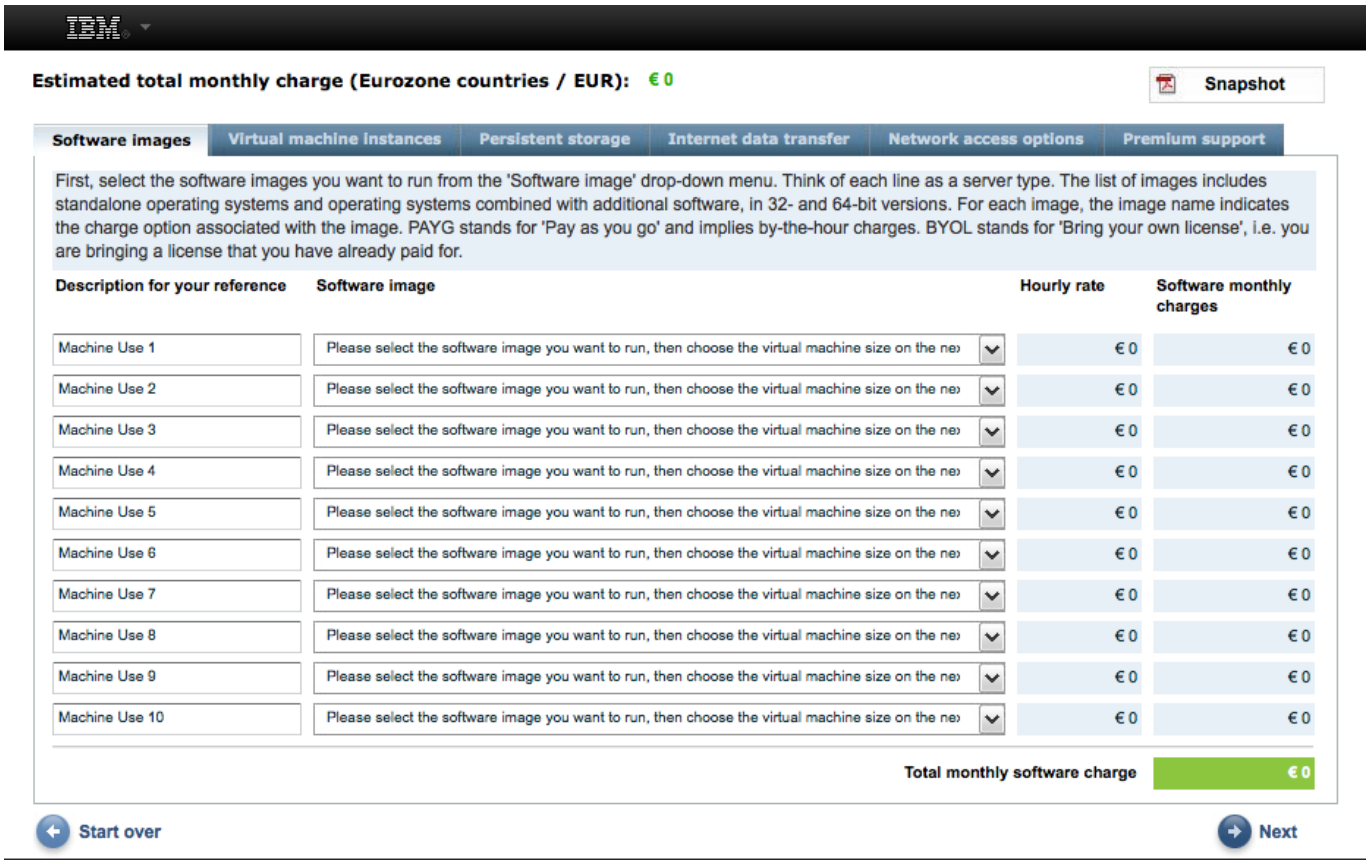

Abb. 9: IBM Smart Cloud Estimator

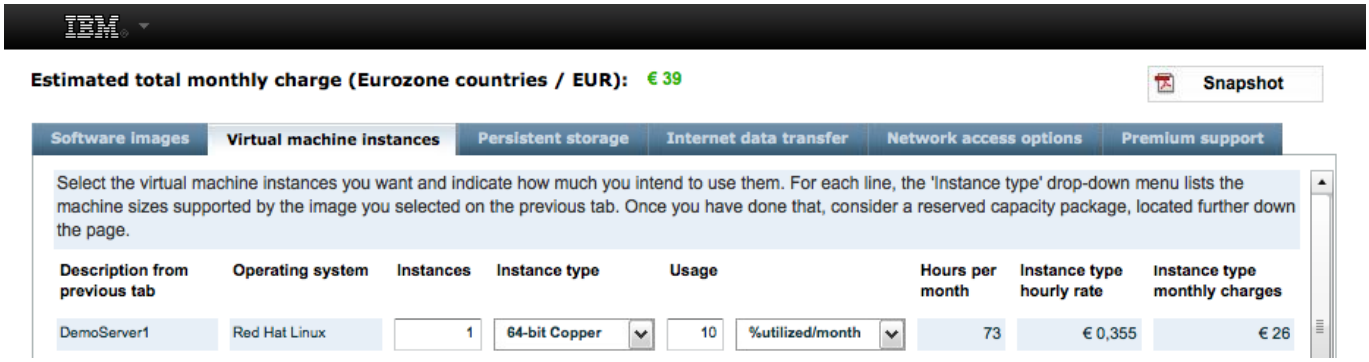

Abb. 10: IBM Smart Cloud Estimator-Kosten

Abschließend möchten wir auf die Cloud-Verwendung der Projektteams innerhalb IBM Global Services (GBS) verweisen. Je nach Kunden- und Vertragssituation ist das Projektteam auf dedizierte Umgebungen angewiesen, welche genau spezifiziert und kalkuliert werden müssen. Bei dem heutigen Zeitdruck sind zuverlässige Angaben bezüglich der Kosten unerlässlich. Damit der Kunde weiß, was ihn die Plattform im Detail kostet, bietet IBM einen umfassenden Monthly Cost Estimator [935] (siehe Abb. 9).

Mit diesem Werkzeug kann man detailliert die Umgebung definieren und mit den nötigen Parametern versehen. Nebst den Softwareimages, wo das gewünschte Betriebssystem mit entsprechender Software ausgewählt wird, sind weitere Angaben betreffend Instanzgröße, persistentem Speicher, Datentransfer sowie Netzwerkzugriff, benötigte IP-Adressen und der gewünschte Support-Level einzutragen (siehe Abb. 10).

Auf Basis dieser Angaben wird ein monatlicher Preis berechnet. Für Kunde und Projektteam sind diese Berechnungen sehr hilfreich und dienen in vielen Fällen als Basis für die Vertragsverhandlungen. ■

Literatur & Links

[OBJ10] OBJEKTspektrum, Ausgabe Cloud_Computing/2010, Titelthema: Cloud Computing: http://www.sigs-datacom.de/fachzeitschriften/objektspektrum/online-themenspecials/artikelansicht.html?show=2879
[Goo] Google trends Virtualisierung-Cloud Computing: http://www.google.de/trends?q=virtualization%2C+cloud+computing&ctab=0&geo=all&date=all&sort=1
[Nis09] The Nist Definition of Cloud Computing“, Oktober 2009: http://www.nist.gov/itl/cloud/upload/cloud-def-v15.pdf
[For] Forrester Definition of Cloud Computing,: http://www.forrester.com/imagesV2/uplmisc/CloudComputingWebinarSlideDeck.pdf
[IDM08] Industry Developments and Models - Global Testing Services: Coming of Age” IDC, 2008 and IBM Internal Reports
[ibm] http://www.ibm.com/cloud-computing/us/en/
[935] http://www-935.ibm.com/services/us/igs/cloud-development/estimator/Tool.htm